AD-A040 637 ILLINOIS UNIV AT URBANA-CHAMPAIGN COORDINATED SCIENCE LAB F/G 9/2
BROWSER A USER ORIENTED INFORMATION RETRIEVAL SYSTEM,(U)
DEC 76 F P CONRAD N00014-75-C-0612
UNCLASSIFIED T-38 NL

1 OF 1
ADA
040637

END
DATE
FILMED
7-77

AD A040637

RERORT T-38 DECEMBER, 1976

12 NW

COORDINATED SCIENCE LABORATORY

BROWSER: A USER ORIENTED INFORMATION RETRIEVAL SYSTEM

FORREST PAUL CONRAD

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

6

BROWSER
A USER ORIENTED INFORMATION RETRIEVAL SYSTEM

by

10 Forrest Paul Conrad

437 049-364

This work was supported by the Office of Naval Research under Contract N00014-75-C-0612

15

11 Dec 76

14 TR-38

12 46p.

097 700

LB

Table of Contents

CHAPTER I

INTRODUCTION

The BROWSER system was designed as a tool to aid in problem isolation, problem measurement and revision monitoring in the maintenance of aircraft by the U.S. Navy. This paper discusses how BROWSER's design accomplishes these goals. Since The overall system would require man-years to complete. ~~Consequently~~ only crucial sections of the code were implemented to ~~as a demonstration~~ of demonstrate the system's feasibility. Since simplicity was a paramount design principle, it is not anticipated that the expansion of the system will encounter extraordinary problems. Note however that the design process is an iterative one. It is expected that the final design of BROWSER will have changed to accommodate unforeseen problems and streamlining techniques.

BROWSER is a complex information retrieval system. The user interacts with the system in real time through a computer terminal. This is to be differentiated from retrieving the data on a real time basis. The data base which BROWSER operates on is so large that a majority of it resides on magnetic tape. Accessing the data base will typically involve the mounting of tapes on a limited number of read/write devices. Information retrieval thus depends on circumstances outside of BROWSER's control. Here then is the first service BROWSER has to offer. Rather than schedule a time when the user and the data are available, BROWSER accepts the user's query, supervises processing of the query and notifies the user upon completion. The period of processing may span several days during which time the user can perform unrelated work. After processing the query, BROWSER notifies the user upon completion.

Real time interaction with the user takes the form of an english dialog. The user can type an english description of his/her query into the system. An analysis of the input is performed. If the system finds ambiguities it cannot resolve or finds that not enough information was specified, it will notify the user. This dialog will continue until the system is satisfied that enough information has been obtained to successfully supervise processing of the query.

The next step depends on how much time is required to answer the query. In cases where the delay is on the order of one minute the user will probably wait. For greater intervals the user will tell the system what action to take when processing is completed.

Entering a particular query is usually only one step in the total procedure which the user employs to answer his/her top level query. A user interested in determining the causes of an aircraft crash will ask many questions about the aircraft's maintenance history before he/she pinpoints the problem. He/she will typically ask a series of questions each of which narrow the realm in which the problem can reside. Following each question he/she will examine (test) the results and determine (branch) the direction he/she feels is most fruitful for succeeding questions. Eventually he/she will converge on an acceptable answer to his/her top level query. BROWSER is designed to aid this process. In addition to the supervision of individual query processing, the system maintains a record of the course of these queries. The user may refer to previous queries and previous query answers when he/she is constructing the current query.

The most sophisticated feature of the BROWSER system is that it allows the user to enter a top level query and then applies a series of its own questions in an attempt to converge on a solution. The top level query must, however, have been predefined within the BROWSER system. The user might simply enter the query "Determine possible causes for the crash of aircraft X." A procedure is followed which determines whether known problems have occurred in the aircraft's maintenance history. After each of the system's questions, it will perform a test to decide if this area requires more processing. If the test results are negative, the system branches to a new question.

BROWSER was designed to parallel the procedure a human would use to isolate a problem and its causes. The procedure which it uses to process a top level query can be thought of as a coded version of the process through which a human would achieve the same results. The difference is that BROWSER can only handle those sequences of questions which can be foreseen. After each question in a sequence, BROWSER performs a test. If the system decides the test can not be applied to the retrieved data, i.e. the question's answer, it has no recourse but to terminate that line of questioning. In a similar situation a human would re-evaluate the data and decide on a new tack.

Many queries would benefit from a merging of the characteristics of man and machine. Many problems fall into categories where creative effort is not necessary and unexpected results are rare; a well-defined technique for solving the problem exists. The structure of other problems is not delineated to the point of providing such a technique. BROWSER can handle

routine problems almost unassisted while its interactive ability gives the user access to all of BROWSER's facilities for cooperative creative problem solving. The system will process a top level query to the fullest extent possible. It is up to the user to use this information and interactively refine the data until he/she reaches an acceptable solution.

BROWSER's capacity to process top level queries on a stand alone basis makes possible one more level of sophistication. Each top level query is in essence a description of a particular problem that is encountered in the maintenance of aircraft. BROWSER has a library of these queries. With supervision from BROWSER, each of these queries can be applied to each aircraft in the data base to determine whether that problem exists for the individual plane. BROWSER received its name because its ultimate goal is to browse through the data base looking for maintenance problems. If the system has no work assigned, it automatically begins searching the data base on its own for problems. Should any be discovered, the user is alerted.

Finally, note the hierarchical structure of BROWSER. Each level builds on the previous level. All the functions of BROWSER use the same data retrieval mechanisms. The design is such that each routine in the system has a unique function. This greatly simplifies updating and maintaining the system. Any change is automatically reflected throughout the system.

CHAPTER II

MODES OF OPERATION

The system design incorporates three modes of operation: interactive, foreground and background. Each mode corresponds to a specific function that BROWSER is performing. Interactive mode means the system is communicating with the user in real time. Foreground mode means that the system is processing user supplied queries. Background mode means the system is browsing through the data base on its own looking for maintenance patterns which may indicate a problem.

The introduction of modes is mostly a device for defining BROWSER's control structure. Each mode will use the same data retrieval routines. The difference between the modes is not which routine is active but how the processing is supervised and what is done with the results. Interactive mode indicates that results are shown directly to the user. Foreground mode results may be queued for the user or may indicate that the next query in the sequence should be processed. Background mode may queue its results, select the next query in the sequence or select a new sequence altogether.

Again, the mode of operation indicates the function the system is performing. In a multi-tasking environment using re-entrant code it may be the case that all three modes are active concurrently. In a different environment it is possible that only one mode at a time can be active. The only requirement is that BROWSER's Supervisor component controls the system in such a manner that each mode can perform its role to a successful completion.

Interactive mode communicates directly with the user at his/her terminal. The PLANES system analyzes the user's natural language input. If PLANES feels it has understood the user correctly, it will pass a machine understandable translation to the BROWSER system. If PLANES is not satisfied, it will return to the user and explain the difficulty. PLANES is a complex system and is documented elsewhere [Waltz (1976)].

Interactive mode allows the user to:

1. Get information concerning the use and limitations of the BROWSER system, e.g. "Can you calculate standard deviations?"

2. Get immediate answers to questions where the data is currently available, e.g. "Who manufactures part x?"

3. Retrieve answers to previously processed questions.

4. Enter a question for later execution if the answer requires extensive processing, e.g. "Determine possible causes for the crash of aircraft X."

5. Direct various system functions, e.g. "Graph the output of the last question.

6. Enter data into one of BROWSER's files, e.g. "Note that part X has a verified manufacturer's defect which results in turbine case ruptures."

When the user enters a question into the system and he/she does not want to wait for the answer or the projected turnaround time is long, the question is entered into a queue. In foreground mode the system selects a question from the queue and determines what is necessary to obtain an answer. The question may only require simple information retrieval. However, the question may require the application of a procedure, i.e. sequence of queries, to obtain an answer. In either case the system

documents the course of execution and keeps track of intermediate results. The final result is queued so the user can examine it at his/her convenience.

Background mode is similar in operation to foreground mode except that the questions it processes do not originate with the user and it does not terminate after a single question is processed to completion. While foreground mode may process "Did aircraft X have any significant failures in the last year?", background mode would apply this question to every aircraft in the data base. Foreground mode would return the answer to the user upon completion. After checking all aircraft and reporting the results, background mode selects another question and applies it to all the aircraft in the data base. Foreground mode will remain active only as long as there are questions in its work queue.

Note that all three modes use the same routines and it is only the parameters that differ. In the remaining portion of this paper the mode of operation will be largely disregarded. Explicit reference to the mode will be made in discussions where its relevant.

CHAPTER III

THE SYSTEM CONFIGURATION

Figure 1 is a schematic representation of BROWSER. This configuration represents conceptual and functional divisions rather than divisions between sections of code. Each component will be described separately.

Although the user is central to the total system design, very little of this influence will be elaborated. The PLANES system is responsible for man-machine interaction. It is a complex system and will not be described here [Waltz (1976)]. It is adequate to say that PLANES is a system which understands a subset of the english language and can translate this input into a machine-understandable paraphrase. The PLANES system has the ability to do simple data retrieval. For queries in which the user can specify all the information needed to locate the data of interest, the PLANES system can do the retrieval. For queries which involve the application of procedures to isolate the data of interest, the BROWSER system must be activated.

The Input/Output Handler is a collection of routines which controls data transfers. It performs such tasks as locating, loading and positioning magnetic tapes. For instance, if the maintenance data for an A-7 with tail number 123456 during June 1971 is needed, the I/O Handler finds the tape and the location of the data on the tape by consulting a directory. It then transfers this data into a temporary disk file so that it can be processed. The performance of these functions by the I/O Handler is not relevant to this paper.

The data base search routines are documented elsewhere [Green 1976]. Some sample search queries will be presented later in this paper with an explanation. This is to give the reader a feeling for the low level routines which BROWSER uses.

A description of the remaining components follows.

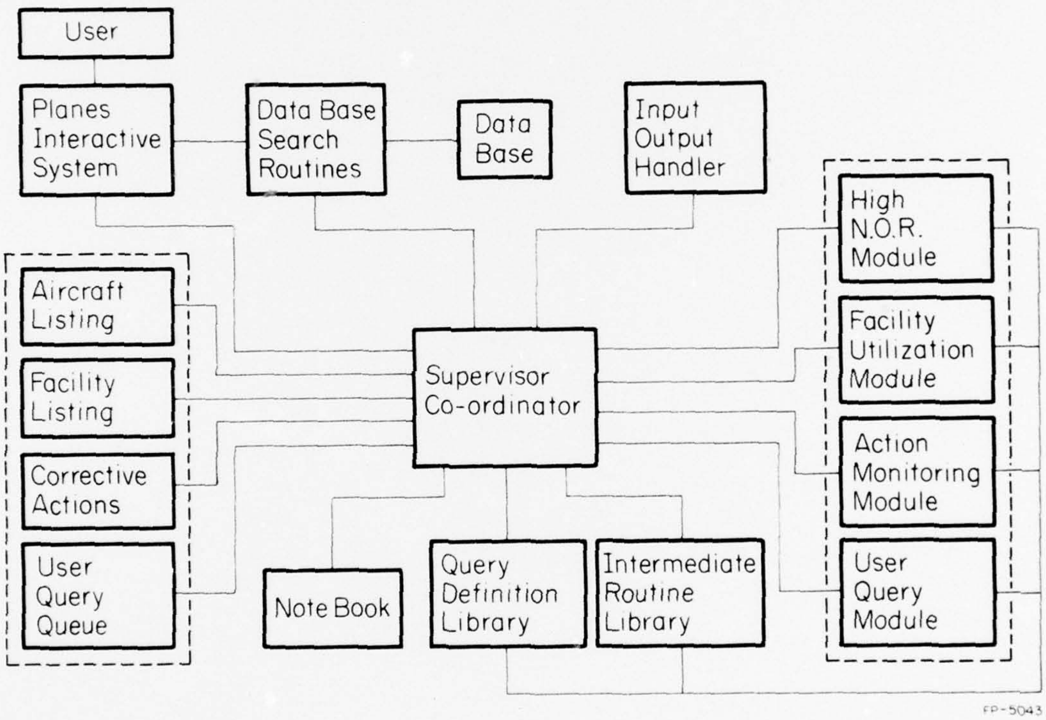

Fig. 1. Schematic description of BROWSER.

CHAPTER IV

THE SUPERVISOR

The Supervisor is the communication center of BROWSER. Virtually all messages sent between system components must pass through it. After examining a message, the Supervisor may alter it before passing it to its destination. There are six functions which the Supervisor performs:

1. Data transfers
2. The setting of system parameters
3. Temporary answer storage
4. Information checking
5. Mode selection
6. Question management

Data transfers involve the modification of system files. When a user enters a question which requires extensive processing, it must be stored in the User Query Queue. To accomplish this, PLANES sends a message (the query) to the Supervisor along with its destination and purpose. The Supervisor will verify that all information for successful processing of the query has been supplied. After making the addition to the queue, the Supervisor updates its directories to reflect the change. Data transfers for the other system components are handled similarly.

The Supervisor is responsible for setting certain system parameters. The user may want a printed copy of the interactive session. For certain questions the user may want to examine a detailed trace of the process which resulted in the question's answer. The user may wish to see some results

displayed in graph rather than table form. In these and other cases, the Supervisor will set parameters (flags) to achieve the desired results. These parameters tell the Supervisor how to perform its role.

Most query processing will involve a substantial turnaround time. BROWSER will work on these queries after the user has left the terminal and will probably finish at a time when the user is not at a terminal. It is the Supervisor's responsibility to store the query, the results and other relevant information. The user can then resume his/her terminal session whenever it is convenient and examine the results.

One of the most interesting and useful features of the Supervisor is information checking. Suppose one of the modules wants to know the not operationally ready time for aircraft X in June. Suppose also that this information is available because a previous query computed it (query results are saved for an indefinite period in the system Notebook). Before passing the data base search query to the appropriate routines, the Supervisor checks to see if this information is available in the Notebook. This is done by comparing the current query against the queries in the Notebook. A match indicates that the information is already available. In many cases the Supervisor can thus reduce the time required to process a request.

The Supervisor is the only system component which has knowledge of the system's mode of operation: interactive, foreground or background. Each other component need only process the information supplied to it. A data base access routine may have been activated from a different mode on each of three successive occasions. It will retrieve the desired data in

each case and pass it to the Supervisor. The Supervisor passes this data to the proper system component and verifies that the component resumes processing with the proper parameters.

Since each system component, other than the Supervisor, functions independently of any other component, it may work on several questions from different operating modes concurrently. The User Query Module operates in foreground mode on questions in the User Query Queue. At some point in the processing of a query the module will need data retrieved from the data base. It may be hours or days before this data is available. The Supervisor will save all the parameters which identify the query being processed, the point where processing was interrupted and the unsatisfied data request. The Supervisor then selects another question from the queue and the User Query Module begins processing the new query. When the data base routines finally return the data for the original query, the Supervisor will save parameters for the second query and have the User Query Module resume processing of the original query. Eventually both queries will be processed to completion but the processing will be overlapped. Note that only the Supervisor is responsible for coordinating the entire system.

An advantage of each system component operating independently is the ease with which processing can be safeguarded. At critical points in the processing of a query the Supervisor can store the parameters which identify the checkpoint. In the event of a system crash or some other error, the Supervisor can restart the process from a checkpoint rather than return to the beginning. When large amounts of processing or data could be lost by system crashes, the restart facility is invaluable.

Since the Supervisor is the only system component which controls other components, it is responsible for deciding when a particular question will be processed. Question management is the most complex feature in the BROWSER system.

Each of the four modules in BROWSER's system configuration can process one question at a time. The goal of query management is to keep the system actively processing questions. Two conditions may arise which interfere with this goal: the module may finish processing a question or the module may issue a request for data. The Supervisor will store the results of a query processed to completion so the user can access it later. If the module requires data from the data base, the Supervisor stores the module's current parameters so that the module may resume processing when data is available. In either case a new question must be selected for the module to process.

The User Query Module only processes questions from the User Query Queue. If the queue is empty the Supervisor has no choice but to leave the User Query Module inactive. A non-empty queue means the Supervisor must select one of the Queue entries to processing. One of the system parameters that the user can set in interactive mode is the priority of each question in the User Query Queue. The Supervisor selects the queue entry with the highest priority as the next question to be processed.

The High NOR Module, the Facility Utilization Module and the Revision Monitoring Module require a more complex scheme for selecting their next question. Each module is composed of questions which fall within its area of expertise and each question is applied to each member of its

domain. The High NOR Module is concerned with mechanical maintenance problems which affect a particular plane or an entire series of aircraft. It will process such questions as "Did plane X miss any inspections last month?" or "When will maintenance costs force wholesale replacement of aircraft series Y?" The question is applied to each plane or aircraft series in the aircraft listing. The Supervisor is responsible for choosing the questions from the High NOR Module and the particular plane or aircraft series. When this module requires a question, the Supervisor must determine if all questions have been applied to the plane or aircraft series currently being processed. If unprocessed questions exist the next question is applied to the current plane or aircraft series. If all questions have been processed, the Supervisor must choose an unexamined plane or aircraft series for processing. This selection is performed on a priority basis. The user can set system parameters such that plane X will be processed before plane Y or that series X will be processed before series Y. In certain cases a query will apply to a subset of the aircraft listing; the Supervisor must ascertain that jet aircraft are not checked for propellor damage. All of these conditions must be taken into account before the High NOR Module can be activated with a new question.

The Facility Utilization Module and the Revision Monitoring Module apply a series of questions to each element of their domains. The Facility Module cycles through its listing of maintenance facilities while the Revision Module cycles through its listing of incorporated revisions. The Supervisor must determine if each module has applied all questions in its sequence to the domain member currently being processed. If unprocessed questions exist, the next question is applied to the current domain member.

If all questions were processed the Supervisor will choose an unprocessed domain member on a priority basis. The user can specify system parameters such that repair facility X will be processed before repair facility Y or that revision X will be processed before revision Y.

As a final note under query management, observe that it is possible to predict which question and domain member will be processed next for each of the four modules. Since this information is known in advance, the Supervisor could conceivably issue requests for the relevant data in advance. When processing reached the question the data would be available.

CHAPTER V

THE MODULES

The U.S. Navy had documented the questions which their data processing department receives, the frequency these questions are asked and the time period within which the answers should be returned [NALDA]. Logiqal and processing similiarities seem to categorize most of the questions into one of three groups. The first category involves the isolation of aircraft and equipment with particularly bad mechanical properties. The High Not Operationally Ready (NOR) Module is concerned with problems in this category. The second category involves the effective use of personnel and repair facilities. The Facility Utilization Module is concerned with problems in this category. The final category involves the monitoring of technical revisions. The questions try to determine whether engineering proposals or other corrective actions improved the situation they were designed to correct. The Revision Monitoring Module is concerned with problems in this category. Together these three modules contain all the questions that the system can answer. However each module is designed to operate in background mode; independent of the user each module will apply a sequence of questions to each member of its respective domain. The User Query Module was necessary so BROWSER could process any question the user submits providing the question is contained in one of the other three modules.

Each module performs a portion of the task that the user performs at the terminal; they provide the control structure which decides the direction the succeeding questions will follow. The High NOR Module contains several questions which deal with problems associated with missed inspections.

Should the first question find that all inspections have been performed, it is pointless to execute any of the remaining question in the area.

The code associated with each question is organized into a functional unit similar to a subroutine. It accepts certain parameters which define the particular job it is to perform. The unit also has knowledge of prerequisites which must be satisfied to perform successfully; temporary files which must exist or calculations which must have been made. If everything is in order, the unit will request that the Supervisor execute the code for the question which is stored in the Query Definition Library. Upon completion the Supervisor will pass the results to the unit. The unit examines the results and tells the Supervisor whether it should:

1. Execute the next unit in sequence
2. Execute another sequence of units
3. Quit because of an unrecoverable error

All units (questions) which operate within a common conceptual area are grouped into a sequence. All units which deal with some aspect of missing inpsections form one sequence. In some sequences it must be true that all previous units have been executed and have returned positive results; each unit builds on and uses the data returned from previous units. In other sequences the units may check independent aspects of the problem relevant to this area. This dependence or independence of units is a major item which must be taken into account when prerequisites are checked.

The User Query Module utilizes the unit and sequence organization of the other three modules to answer the user's question. In concept the

role of the User Module is to copy and execute the units (questions) or sequences which are required to answer the user's query. Successful execution of these copied units or sequences requires that the User Module perform two actions. First, it must be certain that the prerequisites are satisfied before the unit is executed. This may involve execution of other units or the performance of calculations. Second, the User Module may have to modify parameters supplied to the unit. In most cases the units are designed to process a series of aircraft or a group of repair facilities. If the user asks a question about a particular plane or facility, various parameters may need modification so that the unit processes only the item of interest. When satisfied, the User Module requests execution of the question's code in the Query Definition Library. The Supervisor returns the results to the User Module which passes the results to the duplicated unit. The unit will tell the User Module whether the results were positive. With this information the User Module decides which unit to activate next. If the user's question is completely answered, the User Module informs the Supervisor and is given another user query.

The questions which appear in the High NOR Module, the Facility Utilization Module and the Revision Monitoring Module will be presented in English translation. It would take man-years to install and de-bug all the questions and question sequences which appear below. They are presented to give the reader an idea of the type of questions which BROWSER must process. The code associated with each question represents the bulk of the system. It is in this area that most system updating and maintenance will be performed.

The High NOR Module is envisioned as containing:

A. Checking for wholesale degradation of aircraft

1. Does a trend analysis of failure and maintenance rates differ significantly from the corresponding rates of new aircraft?

2. What is the rate of change of failures and maintenance?

3. Does failure or maintenance rates by work unit code change uniformly? (i.e., are some systems wearing at different rates?)

4. Identify aircraft with low/high acceleration of failure/maintenance rates. (i.e., is this a good or poor failure/maintenance history?)

5. How soon will maintenance costs force replacement of this aircraft series?

B. Individual aircraft/component degradation

1. Cull out the highest failure/maintenance rates by work unit code.

2. Isolate problem to specified level. (e.g., part, unit, or subsystem.)

3. Compare against same units in all other aircraft with this configuration.

4. Compare against same units in this series or type of aircraft.

5. Compare maintenance histories of crashed and current aircraft which use this part.

6. Compare inspection schedule with master schedule to determine missed inspections.

7. Compare inspection histories of crashed and current aircraft that use this part.

8. Determine if aircraft had all applicable technical directives installed.

9. Compare technical directive installations on crashed and current aircraft.

10. Did crashed and current aircraft use the same maintenance facilities?

11. Were malfunctions due to improper handling of this aircraft?

12. Was there a high rate of cannibalization for the aircraft?

13. Were there any other user specified conditions present?

The Revision Monitoring Module is envisioned as containing:

1. Compare failure/maintenance rates before and after a corrective action was performed by series.

2. Compare failure/maintenance rates before and after a corrective action was performed across all aircraft incorporating the action.

3. Were the post corrective action failure/maintenance rates significantly better than those predicted by trend analysis without the corrective action?

The Facility Utilization Module is envisioned as containing:

1. Compare the cost of repairing or reworking parts in maintenance facilities to the cost of replacement.

2. Compare the failure/maintenance rates of reworked/repaired assemblies with the rates of new assemblies.

3. Compare the average turnaround time by work unit code across maintenance facilities.

4. Find maintenance facilities with high NOR aircraft.

5. Does a high no defect or no repair rate exist?

6. Find the percent of turnaround due to awaiting parts.

7. Find the number of aircraft serviced or cannibalized by this facility which had no flight hours in the last 30/60/90 days.

CHAPTER VI

THE DOMAINS

Each of the four modules has associated with it a domain or queue of items. Items are selected from the proper domain for processing by the Supervisor at the appropriate time. Each domain element has a processing priority associated with it. In the absence of other information the Supervisor will select the element within the domain which has the highest processing priority. A module will remain active as long as unprocessed items exist in its domain.

The domain associated with the User Query Module is the User Query Queue. When the User Module requires another question for processing, the Supervisor selects the domain element with the highest priority and passes it to the User Module. The User Module searches the other three modules for the unit or sequence which will process the user's question. If none is found, the Supervisor is alerted to the discrepancy. If an appropriate unit or sequence is found, the User Module will supervise execution of the question/s and return the results to the Supervisor when the user's query is processed to completion.

The domain associated with the High NOR Module is the Aircraft Listing. The NOR Module is concerned with mechanical problems associated with aircraft. Since the body of the module contains all the units (questions) related to mechanical problems, the prime parameter needed for successful operation is a specific plane's serial number or the designation of a series of aircraft. The Aircraft Listing includes the

serial number and series number for each plane known to the system. Each unit or sequence within the NOR Module can process a single aircraft. To process a series of aircraft the data for all planes in that series are combined. With minor modifications, the unit can then process this combined data as if it were a single plane. The user can assign a priority to each plane or aircraft series in the Aircraft Listing. Suppose the user has asked BROWSER to compile a list of aircraft which departed from their series average of not operationally ready hours by more than 75 percent. The user could assign a high priority to these planes in the Aircraft Listing because problems are more likely to occur in this group. Note that some aircraft may have exceptionally good maintenance records and it might not be worthwhile to examine them. The user could assign a priority of zero to these aircraft and they would effectively be eliminated in the individual aircraft processing but included in the series processing.

The domain associated with the Facility Utilization Module is the Facility Listing. The Facility Module is concerned with the effective use of personnel and repair facilities. The body of this module contains all the units (questions) relevant to this endeavor. The prime parameter needed for successful processing of facilities is the organization code identifying a specific facility. Should the user wish to examine the overall data for a group of facilities, e.g. all facilities which service A-7's, the data is combined for the chosen facilities. With minor modifications this combined data is processed as if it were from a single facility. The user can assign a priority to each facility. Suppose the user has asked BROWSER to compile a list of the twenty facilities with the highest overall turnaround time for

their repair actions. Since it is likely that certain problems will appear in this group of facilities, the user could assign a high priority to them so they will be processed before other facilities. As in the NOR Module, a priority of zero would delete a facility from individual processing but not from any group processing in which it would normally be included.

The domain associated with the Revision Monitoring Module is the Revision Listing. The Revision Module is concerned with measuring the effectiveness of incorporated corrective actions. The body of this module is comprised of units (questions) which monitor different aspects of corrective actions, the principle parameter required for successful processing of corrective actions is the number which identifies the individual actions. Given this number, the system can retrieve additional information such as the situation this action is to correct, the aircraft this action affects and the date on which this action was performed on a particular plane. The user may wish to examine the effectiveness of several corrective actions combined. Suppose a series of corrective actions were required to rectify a problem with the jet engine used in A-7's. Did the combination of actions significantly reduce the problem? If the date of incorporation of each corrective action is taken into account, the data for a particular group of actions can be combined. This combined data can be processed as if it was a single action. As in the previous two modules, each corrective action can be assigned a priority. The user might want to verify that the problem with a certain model of landing gear is corrected before he/she checks a problem in a backup system.

CHAPTER VII

THE LIBRARIES

The library system is composed of three sections: the Query Definition Library, the Intermediate Routine Library and the Data Base Query Library. The library structure is such that only one copy of a routine exists to perform any given function. Modifications and updates of routines are therefore automatically reflected in all questions which encompass this function.

The libraries are hierarchically organized according to the level of processing its members perform. The Data Base Query Library contains the lowest level routines. The other two libraries can refer to routines which it contains, but the Data Base Library refers to no other library. This library is the only component of BROWSER which manipulates and has direct access to the data base. When a member of the Data Base Library is executed, all information to guide its action must have been specified in detail. The names of files, the target values of particular data fields, a description of the items to be returned and the order in which the results should be sorted must all be given before a data base query can be successfully processed. The query retrieves the requested data and passes it to the Supervisor which passes it to the system component which requested the data.

The middle level in the library hierarchy is the Intermediate Routine Library. Members of this library can refer to data base queries or to other members of the Intermediate Library. Each member of this library is a subroutine which achieves a particular goal, such as locating inspections which plane X missed. During the achievement of this goal several

data base queries may be issued. In some instances other members of the same library may be invoked. The location of missed inspections will require a list of performed inspections and a list of scheduled inspections. The missing inspection routine may call two intermediate routines to create the desired lists and then perform a comparison of the results. The Intermediate Routine returns the results of its processing to the Supervisor which passes it on to a Module or the Query Definition Library depending on where the request originated.

The highest level of the hierarchy is the Query Definition Library. Members of this library can refer to routines in either of the other libraries or to other members of the Query Definition Library. The difference between members of the Query Definition and the Intermediate Libraries is primarily conceptual. The Query Definition Library contains questions that the user typically asks, while the Intermediate Library contains routines for answering these questions. A typical goal for a Definition Library member is to compare the inspection histories of crashed and current aircraft which used part X. A typical goal for the Intermediate Library is to find missed inspections on plane X. Members of the Query Definition Library are subroutines which achieve a desired goal. The attainment of that goal may be very complicated and involve several calls to members of all three libraries. The processed results are passed to the Supervisor which passes them to the module that originated the request.

Each of the four modules is a collection of units. Each unit is code which handles errors and analyzes the results of one question. In operation, each unit performs some initial checking, invokes the appropriate

member of the Query Definition or Intermediate Library, analyses the results returned by the Library member and passes this analysis to the Supervisor. The Supervisor takes action as described previously to sequence and execute questions.

CHAPTER VIII

THE NOTEBOOK

With a Data Base as large as BROWSER's it is highly desirable to avoid, when possible, full data base searches. The Notebook is a device for increasing system efficiency by avoiding duplication of effort. Whenever BROWSER performs an operation on the data base a temporary file is created to store the results of that operation. Each time the system performs an operation on the data base or any component requires attention, the Supervisor is activated to handle the request. Whenever this request is processed successfully, the Supervisor makes an entry in the Notebook describing the query which invoked this action and the location where the results are stored. Should any system component later issue a similar query, the Supervisor will pass on to the component the previously computed results if they are available.

The method used to find previously computed results is a comparison of the current request for data against the entries in the corresponding section of the Notebook. A complete match indicates the query was processed previously and the results are still available. A partial match may be almost as useful, depending on the nature of the mismatch. Suppose the current and partially matched queries are identical except for the time span. If the current query's time period is a subset of the previous query's time period, the data is useable. The Supervisor tells the data base routines to search the appropriate temporary file instead of expending additional time to do a full data base search. In other circumstances the Supervisor might combine several temporary files to provide the required file. Techniques

used in relational data systems can be employed to determine if the file required for the current query could be constructed from available files [Date (1975)]. The data base queries would then search this constructed file rather than the full data base.

When a query is processed to completion the results can assume two forms. The query might be "How many A-7's had bird strike damage in April?" In this case the answer is a single number produced by creating a file with all the aircraft which satisfied the criteria and counting the entries in the file. The Supervisor must save the temporary file as well as the answer. In the second case the query might be "Show me the data on all A-7's which had bird strike damage in April." Since the answer and the temporary file are one and the same, the Supervisor simply saves the temporary file, Entries in the Notebook indicate whether the answer and the file used to produce the answer are available. A query such as "How many A-7's had bird strike damage in the first half of April?" can then be answered from the temporary file used to produce the answer for the first case presented above.

The Notebook is composed of four sections. The first three correspond to entries in the Query Definition Library, the Intermediate Routine Library, and the Data Base Query Library. The fourth section is a listing of aircraft parts which are known to possess undesirable mechanical properties.

The Query Definition Library contains subroutines which achieve a particular goal when they are invoked with specific parameters. If the subroutine is successful in achieving its goal, the Supervisor makes an entry in the query section of the Notebook. This entry consists of the subroutine

name, the parameters which were used in this invocation, the name of the temporary file from which the answer was generated and the answer (if different from the temporary file).

The Intermediate Routine Library contains subroutines which perform a particular function when invoked with specific parameters. If the subroutine is successful, the Supervisor makes an entry in the intermediate section of the Notebook. The entry is the same as entries in the query section of the Notebook.

The Data Base Query Library contains routines that retrieve information from the data base as specified in the parameters with which the routine is invoked. If the routine successfully locates and retrieves the data, the Supervisor makes an entry in the data base section of the Notebook. This entry consists of the routine name, the parameters which were used in this invocation and the name of the temporary file which contains the retrieved data.

The processing of certain questions within the High NOR Module is concerned with identifying aircraft parts which possess undesirable mechanical properties. Whenever a part is found to be unacceptable by the user, the part number and information associated with the part's function are added to the component section of the Notebook. Should the High NOR Module invoke routines to find faulty aircraft parts, the Supervisor will delete any part appearing in the Notebook's component section from the list of parts the module is checking.

Note that the activation of a routine in the Query Definition Library or the Intermediate Routine Library will produce several entries in the Notebook since the routine may invoke several other routines to accomplish its purpose. Although no measurements are available, it is hoped that this

proliferation of temporary files will aid further queries investigating this area; if the user is interested in isolating faulty aircraft components on A-7's, it is hoped that his/her first question will create enough temporary files dealing with A-7's that relatively few of his/her successive queries will require a full data base search. A good file management scheme is required because of the size and number of files generated. Several techniques exist for deciding when a file is no longer useful. The best technique will have to be chosen after the system is implemented and the effect of each technique can accurately be assessed.

A final point concerning the Notebook is its use with hypothetical questions. There are times when it would be useful to process queries with ficticious or modified data. Suppose part X has a verified defect in all units received from the manufacturer. Suppose, further, that the user wished to know the mean time between failures of the aircraft subsystem incorporating part X, assuming part X contained no defect. BROWSER would add part X to the component section of the Notebook with the stipulation that it affect only the current query. Query processing would continue as normal except that part X would not contribute to the mean time between failures. Each time a failure which involved part X was found the Supervisor would cause it to be ignored. The addition, deletion, or modification of items in the Notebook to suit the needs of a hypothetical question would involve minimum effort and would not interfere with normal processing.

CHAPTER IX

THE DATA BASE

The data base encompasses several files which are required to process the variety of questions BROWSER can handle. The complete data base contains data for approximately 5,000 aircraft. The size of the data base is a major variable in estimating how much time is required to process a question. Full scale searching of the data base must be done as efficiently and as seldom as possible if the system is to avoid unreasonably long turnaround times. U.S. Navy documents state that in order for the system to be useful it should process a user's question in the range of four to twenty-four hours [NALDA].

The largest portion of the data base is the file of maintenance data, flight data and readiness data. The maintenance data includes a record of every installation, removal, inspection and maintenance action performed on any plane maintained by the Navy. Here can be found a description of how a particular part failed, the date it failed, how it was discovered and what action was required to correct the problem. The flight data records the start and duration of each flight, the purpose of the flight and the type of landing or takeoff. The readiness data records the time each plane is available to perform its primary mission, the time each plane is available to perform secondary missions, the time spent on unscheduled maintenance and the time the plane is inoperable because it is waiting for parts to arrive. This portion of the data base has been collected for approximately fifteen years and grows on the order of forty million bits of data per month.

Another portion of the data base is a collection of reliability and maintenance reports. Currently these reports are generated by the Navy

to provide a summary, by aircraft series, of failure rates, maintenance man-hours and flight data. This report provides a measurement of the average aircraft. By consulting this report, BROWSER can determine if a particular A-7 differs significantly from the average A-7. BROWSER will generate this report on its own after it is implemented.

The aircraft configuration file specifies the parts which are installed on each plane. If the user wishes to know if plane X is using a Westinghouse transceiver, this file would be consulted.

The inspection file is a master listing of when inspections are to be performed on aircraft. Some inspections are to be performed after a certain number of flight hours and some inspections are performed on a month-to-month basis. This file will usually be used to locate missed inspections.

The command file lists the hierarchy of command. If the user wishes to examine all squadrons in a specific wing, this file lists the organization codes which each wing encompasses.

The parts file is a cross reference for names, numbers and interchangable parts. If a Westinghouse transceiver fails and no other is available, this file will specify a suitable alternative.

The manufacturers file identifies the parts supplied by a particular manufacturer. It will also supply the location of the manufacturer and cost per unit of these parts.

The technical directive file is a list of all engineering change proposals which were incorporated to correct some maintenance problem. This file includes the part or system affected, the aircraft which the correction applies to and a description of the corrective action.

CHAPTER X

A TYPICAL QUESTION

In this section I will endeavor to trace the processing of a user's query. It is hoped that an example of this sort will clarify the interaction of different components in the BROWSER system.

Suppose that at some point in time the user has typed in the question "In what mechanical systems does the A-7 with tail number 123456 differ by more than 75 percent from the A-7 system averages?" The PLANES system receives this input, determines through an analysis that the answer can be found by activating a subroutine named HISTORY-COMPARE. To properly invoke HISTORY-COMPARE, four parameters must be supplied: the tail number of the particular plane, the series of the plane, the threshold percentage and the time period within which the investigation is to be performed. The first three parameters are contained in the user's question while the time period must be determined. Typically the question will be embedded in a larger dialog which the user is having with BROWSER. At some point the time period of interest must have been specified for the processing of a previous question. This time period is stored and is applied to all succeeding requests unless a different time period is explicitly mentioned. If this is the first question in the dialog, the PLANES system will ask the user to supply a time period. In this case it was the only recourse PLANES had. If the user had left out the series (A-7), PLANES could look it up in one of BROWSER's files.

Satisfied that it has understood the user's intent, the PLANES system asks the Supervisor to make an addition to the User Query Queue. This entry would appear as:

```
((Queury-Number 1024)
 (Priority 1)
 (HISTORY-COMPARE 123456 A-7 (Aug1975 Aug1976) .75)
```

The use of parentheses and the general format shown above are dictated by the computer language used to code BROWSER. LISP is an interpretive list processing language whose qualities make its use in Browser desirable. The query number is supplied by the PLANES system as a means of uniquely identifying each question the user submits. The priority is a default value of one since the user did choose to supply a priority.

At some point in time the User Query Module will require another question for processing and the Supervisor will select the above question from the User Query Queue. The User Module locates the HISTORY-COMPARE subroutine within the High NOR Module. Remember that the High NOR Module is composed of units and sequences of units. Each unit is code for initiating and analyzing the results of the single question it deals with. The unit which contains the HISTORY-COMPARE subroutine is a member of a sequence whose overall purpose is the isolation of parts with undesirable mechanical properties. However, this unit has no prerequisites associated with it and can be executed without additional processing. If prerequisites had been present, the User Module would have had to satisfy them before executing the HISTORY-COMPARE unit. The prerequisites might have called for retrieval of data or the execution of other units.

The User Module requests execution of the HISTORY-COMPARE unit. Upon completion of processing, the User Module, not the High NOR Module, will receive the results from the Supervisor. If the NOR Module received the

results, it would assume it had been executing the sequence in which the HISTORY-COMPARE unit is embedded and would call for execution of the next unit in the sequence. This is not the case and such action would result in an error message.

Execution of the HISTORY-COMPARE unit invokes the HISTORY-COMPARE subroutine in the Query Definition Library. This subroutine is a series of steps which must be successfully completed to achieve the subroutine's goal: to locate mechanical systems on plane 123456 which differ from the series' average by 75 percent. The first step is to generate the A-7 series system averages. A routine exists in the Intermediate Routine Library to accomplish this. The HISTORY-COMPARE subroutine requests that the Supervisor execute the intermediate routine. The Supervisor notices that a previous question from another user created a system summary for A-7 aircraft which includes the time period covered by the current request. Since this information is still available in a temporary file, the Supervisor does not activate the intermediate routine but supplies the information directly to the HISTORY-COMPARE subroutine.

The second step is to produce a summary of the performance of plane 123456 as a whole, so the user can decide if the plane's overall performance is average. The intermediate routine COMPILE-PLANE-HISTORY will perform the task. The HISTORY-COMPARE routine requests that the Supervisor execute COMPILE-PLANE-HISTORY. This routine's first step is to retrieve all maintenance data relevant to the current query. To accomplish this it requests execution of the following data base search query:

```
( find all
 (( maint m ))
 (( sum ( maint manhours )))
 ( and ( equal ( maint tail-number ) 123456 )
       ( between ( maint date ) aug1975 aug 1976 )))
```

This is a simplified version of the actual request. An English translation is "Find all occurrences of maintenance actions performed on plane 123456 between the dates of August 1975 and August 1976, and return the total manhours expended on maintenance." The file created will hereafter be referred to as the primary file.

The data base query language is documented elsewhere and will not be described in detail [Green (1976)]. Note however that the data base query can process a file and perform computations on the contents. The result of the computation (in this example the summation of manhours) becomes the answer to the query. This is to be contrasted to the temporary file on which the computations were performed. The Supervisor makes an entry in the system Notebook which is composed of three elements: the data base query which produced the results, the name of the temporary file and the computed answer (if any). If the Supervisor later encounters a data base request which matches or partially matches the query stored in the Notebook, the temporary file may be adequate to supply the data without resorting to a full data base search.

When the data base query has returned the maintenance data in a temporary file and the Supervisor has made an entry in the Notebook then control returns to the COMPILE-PLANE-HISTORY routine. The remaining steps in this routine compute various quantities which describe the contents of the primary file: mean time between failures, mean time between maintenance

actions. These quantities are computed by several additional executions of data base queries. Note that these data base queries do not require any further full scale data base searches because the primary file is sufficient for computation of these quantities.

When the COMPILE-PLANES-HISTORY routine finishes, control is returned to the HISTORY-COMPARE routine. At this point in processing the HISTORY-COMPARE routine has completed the first two steps. It has available the A-7 system averages and a set of descriptors (e.g. mean time between failures) which indicate the overall performance of plane 123456. The third step is to compute a set of descriptors for each mechanical system on the plane. A routine exists in the Intermediate Routine Library which operates in a fashion similar to the COMPILE-PLANE-HISTORY routine except that it computes descriptors for a mechanical system rather than a whole aircraft. The HISTORY-COMPARE subroutine executes the COMPILE-SYSTEM-HISTORY routine for each mechanical system it wants to examine. Since the primary file contains all maintenance data on all of plane 123456's mechanical systems, all data base queries will access the primary file rather than the full data base. Each execution of COMPILE-SYSTEM-HISTORY creates a temporary file which contains maintenance data for one system only. Any further computation on a system will process the appropriate system file.

After completion of its third step HISTORY-COMPARE has available the A-7 system averages, a set of descriptors (e.g., mean time between failures) which indicates the overall performance of plane 123456 and a set of descriptors for each mechanical system on plane 123456. Since all required data is available, HISTORY-COMPARE now initiates the fourth step

which determines if a mechanical system on plane 123456 is significantly different than the series average. A closer examination of the A-7 system averages will show that they are identical in format to the sets of descriptors just computed for plane 123456. In fact, the A-7 system averages were computed by a procedure identical to the one outlined for steps two and three of HISTORY-COMPARE. The only difference is that the primary file contained maintenance data for all mechanical systems on all A-7's instead of all mechanical systems on plane 123456.

HISTORY-COMPARE has two collections of data which are in one-to-one correspondence; one collection describes plane 123456 and one set describes an average A-7 aircraft. The fourth step of HISTORY-COMPARE is to determine if plane 123456's descriptors differ from the A-7 series' descriptors by more than 75 percent. This operation is performed by a routine in the Intermediate Routine Library called DESCRIPTOR-COMPARE. HISTORY-COMPARE executes this routine once for each mechanical system and once for the overall totals. On each execution the routine is passed two sets of descriptors; one set for a system on plane 123456 and one set for the corresponding system in the A-7 series summary. DESCRIPTOR-COMPARE compares each descriptor of each set of descriptors. If plane 123456's descriptor is more than 175 percent of the corresponding descriptor for A-7's, a "+" is prefixed to the descriptor. If plane 123456's descriptor is less than 25 percent of the corresponding descriptor for A-7's, a "-" is prefixed to the descriptor.

The fifth and final step of HISTORY-COMPARE is to arrange its results in a table which will be shown to the user. Each row of the table is a set of descriptors for one mechanical system on plane 123456. A "+" prefixing a

number indicates it was more than 175 percent of the corresponding series descriptor. A "-" prefixing a number indicates it was less than 25 percent of the corresponding series descriptor. The absence of a prefix indicates the number did not differ from the series average by more than 75 percent. This format indicates the numeric value of each descriptor as well as its deviation from the average. The Supervisor passes the result (the table) from HISTORY-COMPARE to the User Query Module. The User Query Module in turn will pass the result to the HISTORY-COMPARE unit in the High NOR Module. The unit determines whether any processing errors occurred and takes appropriate action if any are found. Since none occurred, the unit is satisfied and control is returned to the User Query Module. User Query Module determines that the user's original question does not require the additional processing of any other units in the High NOR Module. If the unit just completed was fulfilling a prerequisite for another unit or the user's original query called for the execution of a sequence of units, then the User Query Module would request execution of the indicated unit. Instead the User Query Module informs the Supervisor that Query 1024 was processed successfully. The Supervisor will store the table which was passed to it by the User Query Module and will select another entry in the User Query Queue for the module to process. When the user returns to the terminal, the PLANES system will inform him/her that his/her previous query was processed. PLANES will then display the original query and present the table which was the query's answer. Depending on the user's evaluation of the information in the table, PLANES will typically receive another query in an attempt to pinpoint the maintenance problem.

CHAPTER XI

CONCLUSION

BROWSER is a user oriented information retrieval system in both operation and design. A good computer software system is one which performs its function efficiently. A successful system is one which operates efficiently and presents the user with a convenient way to achieve his/her goal. The interface between the system and the user is the key to the usefulness of any system. If the user is plagued with opaque error messages, intricate instructions for controlling the system and a general ignorance of the system's operation he/she will avoid contact with the system even if this creates additional work for himself.

The PLANES system provides a natural medium for communication with the user because the interaction is in the user's natural language, English. Even more important, however, is that fact that BROWSER exhibits a degree of intelligence. During a dialog with the user, certain information can be omitted and the BROWSER system will supply it from the context of the dialog or by a simple referral process. Questions can also refer to the results of previous questions. The user can enter questions in any format he desires as long as the BROWSER system can unambiguously determine the user's intention. The goal is to approach the level of conversation the user could expect if he/she were communicating with another human being.

The design of BROWSER is user oriented because it incorporates, whenever possible, the techniques a person would use to achieve the same result. Many procedures internal to BROWSER are coded versions of user

dialogs. BROWSER takes this process one step further by applying these internalized dialogs to data without being prompted to such action by the user. It is hoped that such actions can locate a problem area and the user become involved only when creative or original dialogs must be employed to pinpoint the nature of such problems.

The design of BROWSER took into consideration the attitude with which the user approached the system for information retrieval and general software maintenance. Every effort was made to provide a convenient tool which the user could understand. When all is said and done the user is still BROWSER's most valuable component.

REFERENCES

Date, C. J., <u>An Introduction to Data Base Systems</u>, (Addison-Wesley) 1975

Green, Fred, <u>Implementation of a Query Langauge Based on Relational Calculus</u>, (Master's Thesis, University of Illinois) 1976

"System Data Requirements Determination Report," Naval Air Logistics Data Analysis [NALDA], (Naval Aviation Integrated Logistic Support Center, Maryland) 1974

Waltz, D. L., "Natural Language Access to a Large Data Base," <u>Naval Research Reviews</u>, (Office of Naval Research, Va.) January, 1976